REMARKS

Claims 1-20 remain pending in this application. Claims 1-3, 14, 15 and 18 are amended.

Claim 1 is amended to include the limitation that the architecture is of the type having a plurality of entities generating events, where each event includes system information and context sensitive information, and the additional step of outputting the one or more text strings. Support for these amendments can be found in at least paragraphs [0018], [0023], [0027] and [0034] of the specification.

Claim 2 is amended to include the limitation that the events are filtered to process only events from identified entities. Support for this amendment can be found in at least Table 1.

Claim 3 is amended to include the limitation that the chassis logs include chassis codes formed of two numbers. Support for this amendment can be found in at least paragraph [0027].

Claims 14 and 15 are amended to include the limitation that the one or more cells (claim 14) or processors (claim 15) of the architecture are specified as command line options. Support for these amendments can be found in at least Table 1.

Claim 18 is amended to include the limitation that the architecture is of the type having a plurality of entities generating events, where each event includes system information and context sensitive information, and a computer that includes an extraction tool for extracting the events from the architecture and transforming the events into one or more text strings for output. Support for these amendments can be found in at least paragraphs [0018], [0023], [0027] and [0034] of the specification.

No new matter is added.

Claim Rejections – 35 U.S.C. § 101

Claims 1-20 stand rejected under 35 U.S.C §101. Respectfully we disagree. Nonetheless, claims 1 and 18 are amended to provide further justification in overcoming the §101 rejections. For example, claim 1 is amended to include the feature of outputting the one or more text strings. Thus, claim 1 and dependent claims 2-17 produce a useful and tangible output. Claim 18 is amended to recite a computer includes an interface and that events are transformed to text strings for output. Thus, claim 18 and dependent claims 19 and 20 include tangible items (i.e., computer and interface) that produce tangible output.

Reconsideration of claims 1-20 under the 35 U.S.C §101 rejections is respectfully requested.

Claim Rejections – 35 U.S.C. § 102

Claims 1-4 and 6-20 stand rejected under 35 U.S.C. § 102(e) as being anticipated by US Patent Publication Number US2003/0074607 A1 to Brundridge et al. (hereinafter "Brundridge"). Respectfully, we disagree.

To anticipate a claim, Brundridge must teach every element of the claim and "the identical invention must be shown in as complete detail as contained in the ... claim." *MPEP 2131* citing *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, 2 USPQ2d 1051 (Fed. Cir. 1987) and *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913 (Fed. Cir. 1989). Brundridge does not teach every element of claims 1-4 and 6-20.

Claim 1 recites a method for processing events from electronic architecture, the architecture of the type having a plurality of entities generating the events, including the steps of:

a) extracting the events from the architecture;
b) separating the events according to the entities;
c) transforming the events to one or more text strings; and
d) outputting the one or more text strings;
e) wherein each of the events include system information and context sensitive information.

The architecture of claim 1 has more than one entity generating events. That is to say, the events extracted from the architecture are from more than one entity. In particular step a) recites that the events are extracted from the architecture; thus the extracted events are derived from more than one entity. On the other hand Brundridge, in paragraphs [0025-0026], discloses "an end-user with a problem acknowledges the problem and runs a diagnostic program directed to the particular device" and "as the diagnostic program tests a component or device, the diagnostic program generates diagnostic events for each error it encounters." Thus, Brundridge does not derive events from more than one entity, since the diagnostic program is directed only towards one entity.

In the pending office action of March 6, 2006, the Examiner argues that the parsing of Brundridge is equivalent to extracting events from the architecture. Respectfully, we disagree. The parsing of Brundridge "is performed to locate error strings that are compared to a known value." See Brundridge paragraph [0025]. Compare this with paragraph [0003] of

the specification, which states: "as architecture 10 boots, chassis logs are generated from cells 12 to a service processor 30" and "these chassis logs may be accessed from service processor 30 via a communication link." Thus, in accord with the present claims, events are extracted from the architecture before being separated, transformed and output. The step of extracting is therefore significant. As taught by paragraph [0022] of the immediate specification, an input stream of chassis codes may be from a telnet session to architecture 104. That is, the events are extracted from architecture 104 *prior to* processing of the events. The parsing of Brundridge is clearly not equivalent to extracting events from the architecture as required by claim 1.

Step b) recites that the events are separated according to entities. The Examiner asserts that Brundridge teaches separating events according to the entities since Brundridge, in paragraph [0026], discloses an error log that contains only events for the floppy drive. Respectfully we disagree. Brundridge, paragraphs [0025-0026], discloses "an end-user with a problem acknowledges the problem and runs a diagnostic program directed to the particular device" and that "as the diagnostic program tests a component or device, the diagnostic program generates diagnostic events for each error it encounters." Thus, no separation is required to generate the exemplary log of Brundridge, since only one component or device is tested by the diagnostic program. Brundridge cannot therefore anticipate step b).

As disclosed in paragraph [0027] of the immediate application, chassis codes may take the form of two 64-bit numbers, which are then processed to convert to a text string that is sent to an analyzer associated with the entity from which the chassis code was extracted. Brundridge, however, discloses that "upon detection of string compares, the parsing program assembles the appropriate FAQs to be displayed to the end-user." See Brundridge paragraph [0025]. Specifically, the events of Brundridge are used to select FAQs for display and are not analyzed or translated into a written summary of an identified problem relating to the events. Brundridge does not transform the events into one or more strings and therefore cannot anticipate step c) of claim 1.

Step d) of claim 1 requires that the text strings are outputted. Since Brundridge does not anticipate step c), Brundridge does not generate text strings summarizing the events and therefore cannot anticipate step d).

Claim 1 also recites that events include system information and context sensitive information. Brundridge does not disclose that the events contain system information and context sensitive information. In particular, since the diagnostic program of Brundridge is targeted to only one device, the diagnostics are thus specific to that device and do not contain

system information. See for example, the error log output following paragraph [0026] of Brundridge.

For at least these reasons, Brundridge cannot anticipate claim 1; reconsideration is requested.

Claims 2 - 4 and 6-17 depend from claim 1 and benefit from like argument. However, these claims have other features that patentably distinguish over Brundridge. For example, amended claim 2 recites the step of filtering the events to process only events from identified entities. As described table 1 of the present specification, the options –cell and –proc may be used to select which chassis codes to process and hence provide input to the filtering steps 230 and 232 of FIG. 3. Brundridge does not disclose such selective filtering, or anything close to it; Brundridge merely discards duplicate events. See Brundridge page 3, paragraph [0030]. Brundridge cannot therefore anticipate claim 2.

Amended claim 3 recites extracting chassis logs, separating the chassis logs, and transforming the chassis logs to text strings, wherein the chassis logs include chassis codes formed of two numbers. The Examiner asserts that the error logs of Brundridge are similar to chassis logs of the immediate application. We, again, respectfully disagree. The error codes of Brundridge are simple codes, whereas chassis codes are formed of two numbers (one number detailing system information, one number defining context sensitive information). As taught by paragraph [0027], the chassis codes may take the form of two 64-bit numbers and the processing of the chassis codes includes masking the chassis code with a ccmask before converting it to a hex string. Clearly, the chassis codes are not simple codes as used by Brundridge.

Claim 4 recites coupling a getcc extraction tool to the architecture. As previously argued, chassis codes are extracted from the architecture. As recited by claim 4, this may be accomplished by coupling a getcc extraction tool to the architecture. As taught by paragraphs [0019-0028], the getcc extraction tool extracts chassis codes and processes them based upon configuration settings. Brundridge does not disclose such a tool, and therefore cannot anticipate claim 4. The parsing program written in JAVA of Brundridge does not extract events from an electronic architecture and does not couple to the architecture. As disclosed by Brundridge in paragraph [0028], a program is called to parse an error log, and does not couple to the architecture. For at least this reason, Brundridge cannot anticipate the getcc extraction tool of claim 4.

Claim 6 recites that the architecture is a server, wherein the step of extracting events from the architecture comprises extracting events from the server. In view of claim 1,

Brundridge cannot anticipate claim 6. For example, although Brundridge shows servers within a computer network in FIG. 4, Brundridge does not disclose extracting events from these servers.

Claim 7 recites converting a binary representation of the events to text strings. Brundridge does not extract events from electronic architecture, and does not disclose converting a binary representation of the events to text strings. The events of Brundridge are stored as alpha-numeric characters. See for example the exemplary error log after paragraph [0026] of Brundridge. The Examiner argues that the 'errorlevel' of Brundridge anticipates claim 7. Respectfully, we disagree. Although, in paragraph [0025], Brundridge discloses that the error level condition can be a 0 for pass and a 1 for fail, Brundridge does not disclose *converting* this errorlevel to a string. Therefore, Brundridge cannot anticipate claim 7.

Claim 8 recites analyzing the text strings and producing a human interpretable statement summarizing at least one of the events associated with the text strings. Brundridge does not disclose summarizing at least one of the events associated with the text strings. Instead, Brundridge discloses storing event codes retrieved from the log in an array that is used to build a list of frequently asked questions for display to a user. See Brundridge page 3, paragraph [0029]. This list of frequently asked questions is used to further isolate the problem and suggest a possible solution. See Brundridge page 1, paragraph [0005]. Brundridge does not disclose that the FAQs summarize at least one of the events associated with the text strings. The Examiner argues that the content of the FAQs is the statement summarizing the error events because it provides suggested solutions to the errors. However, we contend that this is conjecture and hindsight by the Examiner; the FAQ solution need not have any mention of the errors at all in order to suggest a solution to the problem. Therefore Brundridge cannot anticipate claim 8.

Claim 9 recites that the entities comprise one or more of firmware, software, processors, architecture monitors, power monitors, cabinet monitors, and I/O drivers. As argued above, Brundridge does not disclose extracting events from more than one entity and therefore cannot anticipate claim 9.

Claim 10 recites controlling one or more steps of extracting, separating and transforming via one or more command line options. As argued above, Brundridge does not disclose extracting events from an electronic architecture. Contrary to the Examiner's assertion, Brundridge does not disclose command line options within paragraph [0028] of page 2. As known in the art, a 'command line option' is an input parameter to a command line program. Although Brundridge discloses a command line program, Brundridge make no

disclosure of command line options; the 'errorlevel' of Brundridge is a returned value, and not a command line option. Brundridge does not disclose that the errorlevel is input as a command line option to a command line program. Therefore, Brundridge cannot anticipate claim 10.

Claim 11 recites controlling one or more steps of extracting, separating and transforming according to one or more configuration files. The Examiner argues that the HTML template file of Brundridge is equivalent to the configuration file of the immediate application. Respectfully, we disagree. As known in the art, HTML templates allow display screens to be created programmatically. The HTML template does not control operation of a program in the same manner as a configuration file; the HTML template facilitates output to screen, for example on a web page. Therefore, Brundridge cannot anticipate claim 11.

Claim 12 recites inputting the command line options via a graphical user interface. As argued above, Brundridge does not disclose command line options and cannot anticipate claim 12.

Claim 13 recites updating the command line options automatically from the architecture. The Examiner argues that paragraph [0028] of Brundridge teaches that "the command line program interacts with the devices to receive updated events and generates 'errorlevel' automatically. However, we do not understand how this anticipates updating the command line options automatically, since, as argued above, the errorlevel of Brundridge is not used a command line option. Brundridge, in paragraph [0028], discloses only that if the errorlevel is a non-zero, a program is called. Therefore, Brundridge cannot anticipate claim 13.

Amended claim 14 recites specifying, as command line options, one or more cells of the architecture, and extracting the events only from the one or more cells. As argued above, Brundridge does not disclose command line options or extracting events from an architecture. Brundridge, for at least these reasons, cannot anticipate claim 14.

Amended claim 15 recites specifying, as command line options, one or more processors of the architecture, and extracting the events only from the one or more processors. As argued above, Brundridge does not disclose extracting events from an architecture, and makes no mention of using command line options to extracting events from specific processors of the architecture (see, e.g., table 1 of the present specification). For at least these reasons, Brundridge cannot anticipate claim 15.

Claim 16 recites saving a log file representative of the events. As argued above, Brundridge does not anticipate claim 1 and therefore for at least this reason cannot anticipate claim 16.

Claim 17 recites transmitting the text strings to one or more analyzers associated with one or more entities and analyzing the text strings at the one or more analyzers. The examiner argues that the user of Brundridge represents an analyzer. However, Brundridge does not disclose that the user analyzes the FAQs output of Brundridge. Paragraph [0057] of Brundridge recites "only the relevant FAQ page is presented to the end-user." Thus, in Brundridge, the FAQ page requires no further analysis. For at least this reason, Brundridge cannot anticipate claim 17.

Amended claim 18 recites a system for processing events from electronic architecture, the architecture of the type having a plurality of entities generating the events, including:

a) a computer including an extraction tool for extracting the events from the architecture, separating the events according to the entities, and transforming the events to one or more text strings for output; and

b) an interface for coupling the extraction tool to one or more of the architecture and a log file storing the events from the architecture;

c) wherein each of the events includes system information and context sensitive information.

As argued above, Brundridge does not disclose extracting events from an architecture having a plurality of entities generating the events, separating events according to the entities, and transforming the events into one or more text strings as required by element a) of claim 18. Brundridge does not disclose an interface for coupling the extraction tool to one or more of the architecture and a log file for storing the events from the architecture as required by step b). The Examiner argues that the diagnostic command line application for returning "errorlevel" from the device to the parsing program, and an error log for storing events from the devices anticipates the interface of element b). However, Brundridge in fact does not disclose an interface nor coupling the extraction tool to either the architecture or a log file. Further, Brundridge does not disclose that each event includes system information and context sensitive information as recited by step c).

For at least these reasons, Brundridge cannot anticipate claim 18. Reconsideration of claim 18 is respectfully requested.

Claims 19 and 20 depend from claim 18 and benefit from like argument. However, these claims have additional features that patentably distinguish over Brundridge. For

example, claim 19 recites that the entities comprise one or more of firmware, software, processors, architecture monitors, power monitors, cabinet monitors, and I/O drivers, wherein the events comprise chassis logs from one or more of the firmware, software, processors, architecture monitors, power monitors, cabinet monitors, and I/O drivers. As argued above, Brundridge does not disclose extracting chassis logs from multiple entities, and therefore cannot anticipate claim 19. Claim 20 recites one or more analyzers coupled to the extraction tool, the analyzers processing the text strings into one or more human interpretable statements summarizing at least one of the events associated with the text strings. As argued above, Brundridge does not disclose one or more analyzers, coupled to the extraction tool, for processing text strings into one or more human interpretable statements summarizing at least one of the events associated with the text strings. Therefore, Brundridge cannot anticipate claim 20.

Reconsideration of claims 19 and 20 is respectfully requested.

Claim Rejections – 35 U.S.C. § 103

When applying 35 U.S.C. §103, the following tenets of patent law must be adhered to:

a) The claimed invention must be considered as a whole;

b) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination;

c) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and

d) Reasonable expectation of success is the standard with which obviousness is determined. MPEP §2141.01, *Hodosh v. Block Drug Co., Inc.*, 786 F.2d 1136, 1134 n.5, 229 U.S.P.Q. 182, 187 n.5 (Fed. Cir. 1986).

In addition, it is respectfully noted that to substantiate a *prima facie* case of obviousness the initial burden rests with the Examiner who must fulfill three requirements. **First**, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references or to combine reference teachings. **Second**, there must be a reasonable expectation of success. **Finally**, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. (emphasis and formatting added) MPEP § 2143, *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Claim 5 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Brundridge in view of U.S. Patent Number 6,269,398 granted to Leong et al. (hereinafter "Leong"). Respectfully Applicants disagree.

As argued above Brundridge does not anticipate claim 1, from which claim 5 depends. Leong discloses a method and system for monitoring remote routers in networks. However, Leong does not disclose extracting events from entities of an electronic architecture and does not make up for the shortfall of Brundridge in anticipating claim 1. Therefore, even when combined, Brundridge and Leong do not render claim 5 obvious (since these patents do not teach the elements of claim 5). Reconsideration of claim 5 is respectfully requested.

Applicants have addressed all issues raised in the Final Office Action dated March 6, 2006, and respectfully solicit a Notice of Allowance for claims 1-20. We believe no fees are currently due; however, if any fee is deemed necessary in connection with this Amendment and Response, please charge Deposit Account No. 08-2025.

Respectfully submitted,

By 

Curtis Vock, Reg. No.: 38,356
LATHROP & GAGE, L.C.
4845 Pearl East Circle, Suite 300
Boulder, Colorado 80301
Tele: (720) 931-3011
Fax: (720) 931-3001